```
DDDDDDDDDDD     EEEEEEEEEEEEEEE  BBBBBBBBBBB      UUU         UUU     GGGGGGGGGGG
DDDDDDDDDDDD    EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUU         UUU     GGGGGGGGGGGG
DDDDDDDDDDDD    EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUU         UUU     GGGGGGGGGGGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG
DDD         DDD EEEEEEEEEEE      BBBBBBBBBBBB     UUU         UUU GGG
DDD         DDD EEEEEEEEEEE      BBBBBBBBBBBB     UUU         UUU GGG
DDD         DDD EEEEEEEEEEE      BBBBBBBBBBBB     UUU         UUU GGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG    GGGGGGGGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG    GGGGGGGGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG    GGGGGGGGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG          GGG
DDD         DDD EEE              BBB         BBB  UUU         UUU GGG          GGG
DDDDDDDDDDDD    EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUUUUUUUUUUUUUU     GGGGGGGGG
DDDDDDDDDDDD    EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUUUUUUUUUUUUUU     GGGGGGGGG
DDDDDDDDDDDD    EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUUUUUUUUUUUUUU     GGGGGGGGG
```

!LEID**DBGCALL

```
DDDDDDDD    BBBBBBBB      GGGGGGGG    CCCCCCCC    AAAAAA    LL          LL
DDDDDDDD    BBBBBBBB      GGGGGGGG    CCCCCCCC    AAAAAA    LL          LL
DD      DD  BB      BB  GG            CC          AA    AA  LL          LL
DD      DD  BB      BB  GG            CC          AA    AA  LL          LL
DD      DD  BB      BB  GG            CC          AA    AA  LL          LL
DD      DD  BB      BB  GG            CC          AA    AA  LL          LL
DD      DD  BBBBBBBB    GG            CC          AA    AA  LL          LL
DD      DD  BBBBBBBB    GG            CC          AA    AA  LL          LL
DD      DD  BB      BB  GG  GGGGGG    CC          AAAAAAAAAA  LL          LL
DD      DD  BB      BB  GG  GGGGGG    CC          AAAAAAAAAA  LL          LL
DD      DD  BB      BB  GG      GG    CC          AA    AA  LL          LL          ....
DD      DD  BB      BB  GG      GG    CC          AA    AA  LL          LL          ....
DDDDDDDD    BBBBBBBB      GGGGGG      CCCCCCCC    AA    AA  LLLLLLLLLL  LLLLLLLLLL  ....
DDDDDDDD    BBBBBBBB      GGGGGG      CCCCCCCC    AA    AA  LLLLLLLLLL  LLLLLLLLLL  ....
```

```
LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
;     1          0001  0 MODULE DBGCALL(IDENT = 'V04-000') =
;     2          0002  0
;     3          0003  1 BEGIN
;     4          0004  1
;     5          0005  1 !
;     6          0006  1 !**************************************************************************
;     7          0007  1 !*                                                                        *
;     8          0008  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                               *
;     9          0009  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                 *
;    10          0010  1 !*  ALL RIGHTS RESERVED.                                                   *
;    11          0011  1 !*                                                                        *
;    12          0012  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    13          0013  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    14          0014  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    15          0015  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    16          0016  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    17          0017  1 !*  TRANSFERRED.                                                           *
;    18          0018  1 !*                                                                        *
;    19          0019  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    20          0020  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    21          0021  1 !*  CORPORATION.                                                           *
;    22          0022  1 !*                                                                        *
;    23          0023  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
;    24          0024  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
;    25          0025  1 !*                                                                        *
;    26          0026  1 !*                                                                        *
;    27          0027  1 !**************************************************************************
;    28          0028  1 !
;    29          0029  1 ! WRITTEN BY
;    30          0030  1 !       Ping Sager      Oct. 1982
;    31          0031  1 !
;    32          0032  1 ! MODULE FUNCTION
;    33          0033  1 !       This module contains the parse and execution routines to support the
;    34          0034  1 !       CALL command.  Parsing is done by means of ATN's.  A command execution
;    35          0035  1 !       tree is constructed during parsing.  This tree is passed as input to
;    36          0036  1 !       the command execution network.  The CALL command allows the user to
;    37          0037  1 !       call a subroutine from DEBUG, have it execute, and then view its
;    38          0038  1 !       return value.  The CALL command is language independent, and does not
;    39          0039  1 !       understand the argument passing conventions used by the various
;    40          0040  1 !       languages.  Hence the %ADDR, %REF, %VAL, and %DESCR constructs are
;    41          0041  1 !       are provided by DEBUG.  %ADDR allows the user to specify an address
;    42          0042  1 !       expression and pass in the value of that expression as the parameter,
;    43          0043  1 !       %REF allows the user to specify a language expresion and pass in
;    44          0044  1 !       the address of the expression result (pass by reference), %VAL allows
;    45          0045  1 !       the user to specify a language expresion and pass in the value of the
;    46          0046  1 !       expression as an immediate parameter, and %DESCR allows the user to
;    47          0047  1 !       specify a language expression and pass in the expression result by
;    48          0048  1 !       VAX standard descriptor.  %ADDR, %REF, %VAL, and %DESCR are treated
;    49          0049  1 !       as keywords (not abbreviations), so the user must enter them with
;    50          0050  1 !       those exact spellings.
;    51          0051  1 !
;    52          0052  1
;    53          0053  1 REQUIRE 'SRC$:DBGPROLOG.REQ';
;    54          0187  1
;    55          0188  1 FORWARD ROUTINE
;    56          0189  1     DBG$NEXECUTE_CALL,                  ! Command execution network
;    57          0190  1     DBG$NPARSE_CALL;                    ! Parse network
```

```
;    59          0191  1 EXTERNAL ROUTINE
;    60          0192  1     DBG$GET_MEMORY,                 ! Get a memory block
;    61          0193  1     DBG$GET_TEMPMEM,                ! Get temporary memory
;    62          0194  1     DBG$MAKE_VMS_DESC,              ! Convert Primary Descriptor to
;    63          0195  1                                     !      VAX standard descriptor
;    64          0196  1     DBG$NCOPY_DESC,                 ! Copies primary and value descriptor
;    65          0197  1     DBG$NMATCH,                     ! Counted string matching routine
;    66          0198  1                                     !      for parsing
;    67          0199  1     DBG$NNEXT_WORD,                 ! Isolate next word of input for
;    68          0200  1                                     !      syntax errors
;    69          0201  1     DBG$NPARSE_ADDRESS,             ! Address Expression Parser
;    70          0202  1     DBG$NPARSE_EXPRESSION,          ! Language Expression Parser
;    71          0203  1     DBG$NSAVE_STRING,               ! Store a ASCIC string from input buffer
;    72          0204  1     DBG$PRIM_TO_ADDR;               ! Convert Primary Descriptor to
;    73          0205  1                                     !      Value Descriptor containing
;    74          0206  1                                     !      address of descriptor
;    75          0207  1
;    76          0208  1 EXTERNAL
;    77          0209  1     DBG$GB_TAKE_CMD: BYTE,          ! Flag that says take further commands
;    78          0210  1     DBG$PSEUDO_PROG,                ! Address of phony user code
;    79          0211  1     DBG$RUNFRAME: BLOCK[,BYTE],     ! Current user runframe context
;    80          0212  1     DBG$GB_UNHANDLED_EXC: VECTOR[10,BYTE];! Flags set to TRUE on an unhandled
;    81          0213  1                                     ! exception.
;    82          0214  1
;    83          0215  1 GLOBAL
;    84          0216  1     DBG$GL_CALL_CONTEXT: INITIAL(0),! Used for Bound Procedure
;    85          0217  1     DBG$GB_CALL_NORMAL_RET: BYTE    ! Normal return from CALL command flag
;    86          0218  1                       INITIAL(0);   !      used to suppress regeneration
;    87          0219  1                                     !      of screen displays on normal
;    88          0220  1                                     !      return from the CALL command
;    89          0221  1                                     ! This flag can have these values:
;    90          0222  1                                     !      0 = Not in a CALL command
;    91          0223  1                                     !      1 = In a CALL command, but call
;    92          0224  1                                     !          has not returned normally
;    93          0225  1                                     !      2 = CALL command just returned
;    94          0226  1                                     !          normally without intervening
;    95          0227  1                                     !          breaks or exceptions
;    96          0228  1
;    97          0229  1 OWN
;    98          0230  1     SAVE_CALL_CONTEXT: INITIAL(0);
```

```
;  100     0231  1 GLOBAL ROUTINE DBG$NEXECUTE_CALL(VERB_NODE, MESSAGE_VECT) =
;  101     0232  1 !
;  102     0233  1 ! FUNCTION
;  103     0234  1 !       This routine accepts a command execution tree as input and performs the
;  104     0235  1 !       semantic actions associated with the CALL command.  This routine
;  105     0236  1 !       builds a standard VAX call frame for the user-specified called-address.
;  106     0237  1 !
;  107     0238  1 !       Adverb Node in the command execution tree specifies the called-address.
;  108     0239  1 !       The arguments to the called-address are found in the Noun Nodes in the
;  109     0240  1 !       command execution tree.  The arguments are counted, and if any exist,
;  110     0241  1 !       a standard VAX call frame argument list is constructed.  The the
;  111     0242  1 !       called-address is called via a CALLG instruction, and the returned
;  112     0243  1 !       value from the CALLG is displayed.
;  113     0244  1 !
;  114     0245  1 ! INPUTS
;  115     0246  1 !       VERB_NODE       - A longword containing the address of the verb
;  116     0247  1 !                         node of the command execution tree. (CALL)
;  117     0248  1 !
;  118     0249  1 !       MESSAGE_VECT    - The address of a longword to contain the address
;  119     0250  1 !                         of a standard message argument vector on errors.
;  120     0251  1 !
;  121     0252  1 ! OUTPUTS
;  122     0253  1 !       STS$K_SUCCESS (1)       - Success.  The parsed command was executed.
;  123     0254  1 !
;  124     0255  1 !       STS$K_SEVERE  (4)       - Failure.  The command could not be executed.
;  125     0256  1 !
;  126     0257  1
;  127     0258  2     BEGIN
;  128     0259  2
;  129     0260  2     MAP
;  130     0261  2         VERB_NODE: REF DBG$VERB_NODE;   ! Pointer to the Verb Node
;  131     0262  2
;  132     0263  2     LOCAL
;  133     0264  2         ADVERB_NODE: REF DBG$ADVERB_NODE,! Pointer to the Adverb Node
;  134     0265  2         ARG_LIST_PTR: REF VECTOR[,LONG],! Pointer to argument list
;  135     0266  2         AST_FLAG,                       ! TRUE for CALL/AST
;  136     0267  2         BUF: REF VECTOR[,BYTE],         ! Pointer to ASCIC string
;  137     0268  2         CALARG_PERMEM: REF VECTOR[,LONG],! Pointer to a vector of memory useage
;  138     0269  2                                         !    pointers
;  139     0270  2         CALL_ADDRESS,                   ! User specified Call-Address
;  140     0271  2         I,                              ! Index to the argument
;  141     0272  2         NOUN_NODE: REF DBG$NOUN_NODE,   ! Pointer to the Noun Node
;  142     0273  2         SAVED_RUNFRAME: REF BLOCK[,BYTE],! Pointer to saved runframe context
;  143     0274  2         VALUE_DESC: REF DBG$VALDESC;    ! Pointer to Value Descriptor
;  144     0275  2
;  145     0276  2
;  146     0277  2     LITERAL
;  147     0278  2         STOCK_USER_PSL = %X'03C00000';  ! Standard user PSL value
;  148     0279  2
;  149     0280  2     BUILTIN
;  150     0281  2         PROBER;
;  151     0282  2
;  152     0283  2
;  153     0284  2
;  154     0285  2     ! Recover the flag that says whether we are to enable ASTs during
;  155     0286  2     ! the call.
;  156     0287  2     !
```

```
;  157     0288  2      AST_FLAG = .VERB_NODE[DBG$B_VERB_COMPOSITE];
;  158     0289  2
;  159     0290  2      ! Recover the routine address to call.  If the address is given by a
;  160     0291  2      ! Primary Descriptor, convert it to a Value Descriptor and get the
;  161     0292  2      ! address of the routine to call from that descriptor.
;  162     0293  2      !
;  163     0294  2      ADVERB_NODE = .VERB_NODE[DBG$L_VERB_ADVERB_PTR];
;  164     0295  2      VALUE_DESC = .ADVERB_NODE[DBG$L_ADVERB_VALUE];
;  165     0296  2      IF .VALUE_DESC[DBG$B_DHDR_TYPE] EQL DBG$K_PRIMARY_DESC
;  166     0297  2      THEN
;  167     0298  3          BEGIN
;  168     0299  3          IF NOT DBG$PRIM_TO_ADDR(.VALUE_DESC, DSC$K_DTYPE_L, VALUE_DESC)
;  169     0300  3          THEN
;  170     0301  3              $DBG_ERROR('DBGCALL\DBG$NEXECUTE_CALL 10');
;  171     0302  3
;  172     0303  3          CALL_ADDRESS = ..VALUE_DESC[DBG$L_VALUE_POINTER];
;  173     0304  3          END
;  174     0305  3
;  175     0306  3
;  176     0307  3      ! If the address to call is given by a Value Descriptor in the first place,
;  177     0308  3      ! get it from that descriptor right away.
;  178     0309  3      !
;  179     0310  2      ELSE
;  180     0311  3          BEGIN
;  181     0312  3          IF .VALUE_DESC[DBG$B_DHDR_TYPE] NEQ DBG$K_V_VALUE_DESC
;  182     0313  3          THEN
;  183     0314  3              $DBG_ERROR('DBGCALL\DBG$NEXECUTE_CALL 20');
;  184     0315  3
;  185     0316  3          CALL_ADDRESS = .VALUE_DESC[DBG$L_VALUE_POINTER];
;  186     0317  3          END;
;  187     0318  2
;  188     0319  2
;  189     0320  2      ! Check for read access to the user specified call address.
;  190     0321  2      !
;  191     0322  2      IF NOT PROBER(%REF(0), %REF(1), .CALL_ADDRESS)
;  192     0323  2      THEN
;  193     0324  2          SIGNAL(DBG$_BADSTARTPC, 1, .CALL_ADDRESS);
;  194     0325  2
;  195     0326  2
;  196     0327  2      ! Allocate spaces for Argument list.
;  197     0328  2      !
;  198     0329  2      ARG_LIST_PTR = DBG$GET_MEMORY(.ADVERB_NODE[DBG$B_ADVERB_LITERAL] + 1);
;  199     0330  2      CALARG_PERMEM = 0;
;  200     0331  2      IF .ADVERB_NODE[DBG$B_ADVERB_LITERAL] NEQ 0
;  201     0332  2      THEN
;  202     0333  2          CALARG_PERMEM = DBG$GET_MEMORY(.ADVERB_NODE[DBG$B_ADVERB_LITERAL]);
;  203     0334  2
;  204     0335  2
;  205     0336  2      ! Construct the Argument List.
;  206     0337  2      !
;  207     0338  2      I = 0;
;  208     0339  2      ARG_LIST_PTR[.I] = .ADVERB_NODE[DBG$B_ADVERB_LITERAL];
;  209     0340  2      NOUN_NODE = .VERB_NODE[DBG$L_VERB_OBJECT_PTR];
;  210     0341  2      WHILE TRUE DO
;  211     0342  3          BEGIN
;  212     0343  3          IF .NOUN_NODE EQL 0 THEN EXITLOOP;
;  213     0344  3          VALUE_DESC = .NOUN_NODE[DBG$L_NOUN_VALUE];
```

```
;  214     0345  3          BUF = .NOUN_NODE[DBG$L_NOUN_VALUE2];
;  215     0346  3          I = .I+1;
;  216     0347  3          SELECTONE TRUE OF
;  217     0348  3              SET
;  218     0349  3              [CH$EQL(5, BUF[1], 5, UPLIT BYTE('%ADDR'))]:
;  219     0350  4                  BEGIN
;  220     0351  4                  IF .VALUE_DESC[DBG$B_DHDR_TYPE] EQL DBG$K_PRIMARY_DESC
;  221     0352  4                  THEN
;  222     0353  5                      BEGIN
;  223     0354  5                      IF NOT DBG$PRIM_TO_ADDR(.VALUE_DESC, DSC$K_DTYPE_L, VALUE_DESC)
;  224     0355  5                      THEN
;  225     0356  5                          $DBG_ERROR('DBGCALL\DBGNEXECUTE_CALL, prim to addr failed');
;  226     0357  5
;  227     0358  5                      ARG_LIST_PTR[.I] = ..VALUE_DESC[DBG$L_VALUE_POINTER];
;  228     0359  5                      END
;  229     0360  5
;  230     0361  4                  ELSE
;  231     0362  5                      BEGIN
;  232     0363  5                      IF .VALUE_DESC[DBG$B_DHDR_TYPE] EQL DBG$K_V_VALUE_DESC
;  233     0364  5                      THEN
;  234     0365  6                          BEGIN
;  235     0366  6                          ARG_LIST_PTR[.I] = .VALUE_DESC[DBG$L_VALUE_POINTER];
;  236     0367  6                          END
;  237     0368  6
;  238     0369  5                      ELSE
;  239     0370  6                          BEGIN
;  240     0371  6                          $DBG_ERROR('DBGCALL\DBG$NEXECUTE_CALL, invalid addr. desc.');
;  241     0372  5                          END;
;  242     0373  5
;  243     0374  4                      END;
;  244     0375  4
;  245     0376  3                  END;
;  246     0377  3
;  247     0378  3              [CH$EQL(6, BUF[1], 6, UPLIT BYTE('%DESCR'))]:
;  248     0379  4                  BEGIN
;  249     0380  4                  IF .VALUE_DESC[DBG$B_DHDR_TYPE] EQL DBG$K_V_VALUE_DESC OR
;  250     0381  4                     .VALUE_DESC[DBG$B_DHDR_TYPE] EQL DBG$K_VALUE_DESC
;  251     0382  4                  THEN
;  252     0383  5                      BEGIN
;  253     0384  5                      DBG$NCOPY_DESC(.VALUE_DESC, VALUE_DESC);
;  254     0385  5                      ARG_LIST_PTR[.I] = VALUE_DESC[DBG$A_VALUE_VMSDESC];
;  255     0386  5                      CALARG_PERMEM[.I - 1] = .VALUE_DESC;
;  256     0387  5                      END
;  257     0388  5
;  258     0389  4                  ELSE
;  259     0390  4                      $DBG_ERROR('DBGCALL\DBG$NEXECUTE invalid val. desc.');
;  260     0391  4
;  261     0392  3                  END;
;  262     0393  3
;  263     0394  3              [CH$EQL(4, BUF[1], 4, UPLIT BYTE('%REF'))]:
;  264     0395  4                  BEGIN
;  265     0396  4                  IF .VALUE_DESC[DBG$B_DHDR_TYPE] EQL DBG$K_V_VALUE_DESC
;  266     0397  4                  THEN
;  267     0398  4                      ARG_LIST_PTR[.I] = .VALUE_DESC[DBG$L_VALUE_POINTER]
;  268     0399  4                  ELSE
;  269     0400  4                      IF .VALUE_DESC[DBG$B_DHDR_TYPE] EQL DBG$K_VALUE_DESC
;  270     0401  4                      THEN
```

```
;  271     0402  5                    BEGIN
;  272     0403  5                    DBG$NCOPY_DESC(.VALUE_DESC, VALUE_DESC);
;  273     0404  5                    ARG_LIST_PTR[.I] = .VALUE_DESC[DBG$L_VALUE_POINTER];
;  274     0405  5                    CALARG_PERMEM[.I - 1] = .VALUE_DESC;
;  275     0406  5                    END
;  276     0407  5
;  277     0408  4                ELSE
;  278     0409  4                    $DBG_ERROR('DBGCALL\DBG$NEXECUTE_CALL, invalid val. desc');
;  279     0410  4
;  280     0411  3                END;
;  281     0412  3
;  282     0413  3            [CH$EQL(4, BUF[1], 4, UPLIT BYTE('%VAL'))]:
;  283     0414  4                BEGIN
;  284     0415  4                IF .VALUE_DESC[DBG$B_DHDR_TYPE] EQL DBG$K_VALUE_DESC
;  285     0416  4                THEN
;  286     0417  5                    BEGIN
;  287     0418  5                    ARG_LIST_PTR[.I] = ..VALUE_DESC[DBG$L_VALUE_POINTER];
;  288     0419  5                    IF .VALUE_DESC[DBG$B_VALUE_DTYPE] EQL DSC$K_DTYPE_V OR
;  289     0420  5                       .VALUE_DESC[DBG$B_VALUE_DTYPE] EQL DSC$K_DTYPE_VU
;  290     0421  5                    THEN
;  291     0422  6                        BEGIN
;  292     0423  6                        IF .VALUE_DESC[DBG$W_VALUE_LENGTH] GTR 32 ! bits
;  293     0424  6                        THEN
;  294     0425  6                            SIGNAL(DBG$_SIZETRUNC);
;  295     0426  6                        END
;  296     0427  6
;  297     0428  5                    ELSE
;  298     0429  5                        IF .VALUE_DESC[DBG$B_VALUE_DTYPE] EQL DSC$K_DTYPE_P
;  299     0430  5                        THEN
;  300     0431  6                            BEGIN
;  301     0432  6                            IF .VALUE_DESC[DBG$W_VALUE_LENGTH] GTR 8 ! digits
;  302     0433  6                            THEN
;  303     0434  6                                SIGNAL(DBG$_SIZETRUNC);
;  304     0435  6                            END
;  305     0436  6
;  306     0437  5                        ELSE
;  307     0438  5                            IF .VALUE_DESC[DBG$W_VALUE_LENGTH] GTR 4 ! bytes
;  308     0439  5                            THEN
;  309     0440  5                                SIGNAL(DBG$_SIZETRUNC);
;  310     0441  5
;  311     0442  5                    END
;  312     0443  5
;  313     0444  4                ELSE
;  314     0445  4                    $DBG_ERROR('DBGCALL\DBG$NEXECUTE_CALL, invalid val. desc');
;  315     0446  4
;  316     0447  3                END;
;  317     0448  3
;  318     0449  3            TES;
;  319     0450  3
;  320     0451  3        NOUN_NODE = .NOUN_NODE[DBG$L_NOUN_LINK];
;  321     0452  3
;  322     0453  2        END;                    ! End of WHILE constructing argument list.
;  323     0454  2
;  324     0455  2
;  325     0456  2    ! Save the current run frame context.  Keep the current register
;  326     0457  2    ! contents, set user PC to the special routine DBG$PSEUDO_PROG
;  327     0458  2    ! in DBGSTART that will call the user-specified call-address.
```

```
;  328          0459  2      ! and clear all flags.
;  329          0460  2      !
;  330          0461  2      SAVED_RUNFRAME = DBG$GET_MEMORY((DBG$K_RUNFR_LEN + 3) / %UPVAL);
;  331          0462  2      CH$MOVE(DBG$K_RUNFR_LEN, DBG$RUNFRAME[0,0,0,0], .SAVED_RUNFRAME);
;  332          0463  2      DBG$RUNFRAME[DBG$L_NEXT_LINK] = .SAVED_RUNFRAME;
;  333          0464  2      DBG$RUNFRAME[DBG$L_USER_PC] = DBG$PSEUDO_PROG;
;  334          0465  2      DBG$RUNFRAME[DBG$L_USER_PSL] = STOCK_USER_PSL;
;  335          0466  2      CH$FILL(0,
;  336          0467  2          DBG$RUNFRAME[DBG$K_RUNFR_LEN,0,0,0] - DBG$RUNFRAME[DBG$W_RUN_STAT],
;  337          0468  2          CH$PTR(DBG$RUNFRAME[DBG$W_RUN_STAT]));
;  338          0469  2      IF .AST_FLAG
;  339          0470  2      THEN
;  340          0471  2          DBG$RUNFRAME[DBG$V_ENAB_AST] = .SAVED_RUNFRAME[DBG$V_ENAB_AST];
;  341          0472  2      DBG$RUNFRAME[DBG$L_FRAME_PTR] = .ARG_LIST_PTR;
;  342          0473  2      DBG$RUNFRAME[DBG$L_CALL_ADDR] = .CAL[_ADDRESS;
;  343          0474  2      DBG$RUNFRAME[DBG$L_SAVE_FLD] = .CALARG_PERMEM;
;  344          0475  2      DBG$RUNFRAME[DBG$L_USER_R1] = .SAVE_CA[L_CONTEXT;
;  345          0476  2
;  346          0477  2
;  347          0478  2      ! Also "push" the stack of flags saying whether an unhandled exception
;  348          0479  2      ! has been encountered. The way this works is that we have a byte
;  349          0480  2      ! vector called DBG$GB_UNHANDLED_EXC. If a serious error gets to
;  350          0481  2      ! our final handler, then DBG$GB_UNHANDLED_EXC[0] gets set to 1
;  351          0482  2      ! in DBGSTART. In DBGSTEPGO, this byte is tested when we see a
;  352          0483  2      ! STEP or GO, and an informational is signalled.
;  353          0484  2      ! The only complication is that we need to stack these flags
;  354          0485  2      ! for CALL. This is what we do here. This code assumes we will
;  355          0486  2      ! not get calls more than 10 levels deep.
;  356          0487  2      !
;  357          0488  2      DECR I FROM 9 TO 1 DO
;  358          0489  2          DBG$GB_UNHANDLED_EXC[.I] = .DBG$GB_UNHANDLED_EXC[.I-1];
;  359          0490  2      DBG$GB_UNHANDLED_EXC[0] = 0;
;  360          0491  2
;  361          0492  2
;  362          0493  2      ! Set flag saying that we are leaving DEBUG through a CALL command, turn
;  363          0494  2      ! off taking commands from the user, and return successfully.
;  364          0495  2      !
;  365          0496  2      DBG$GB_CALL_NORMAL_RET = 1;
;  366          0497  2      DBG$GB_TAKE_CMD = FALSE;
;  367          0498  2      RETURN STS$K_SUCCESS;
;  368          0499  1      END;


                                                                .TITLE  DBGCALL
                                                                .IDENT  \V04-000\

                                                                .PSECT  DBG$PLIT,NOWRT,  SHR,  PIC,0

45 4E 24 47 42 44 5C 4C 4C 41 43 47 42 44 1C  00000 P.AAA:  .ASCII  <28>\DBGCALL\<92>\DBG$NEXECUTE_CALL 10\      ;
   30 31 20 4C 4C 41 43 5F 45 54 55 43 45 58  0000F                                                            ;
45 4E 24 47 42 44 5C 4C 4C 41 43 47 42 44 1C  0001D P.AAB:  .ASCII  <28>\DBGCALL\<92>\DBG$NEXECUTE_CALL 20\      ;
   30 32 20 4C 4C 41 43 5F 45 54 55 43 45 58  0002C                                                            ;
                              52 44 44 41 25  0003A P.AAC:  .ASCII  \%ADDR\                                     ;
58 45 4E 47 42 44 5C 4C 4C 41 43 47 42 44 2D  0003F P.AAD:  .ASCII  \-DBGCALL\<92>\DBGNEXECUTE_CALL, prim to\   ;
69 72 70 20 2C 4C 4C 41 43 5F 45 54 55 43 45  0004E                                                            ;
                        6F 74 20 6D           0005D                                                            ;
      64 65 6C 69 61 66 20 72 64 64 61 20     00061          .ASCII  \ addr failed\                            ;
```

```
45 4E 24 47 42 44 5C 4C 4C 41 43 47 42 44 2E 0006D P.AAE:  .ASCII  \.DBGCALL\<92>\DBG$NEXECUTE_CALL, invali\    ;
6E 69 20 2C 4C 4C 41 43 5F 45 54 55 43 45 58 0007C                                                          ;
                                    69 6C 61 76 0008B                                                          ;
      2E 63 73 65 64 20 2E 72 64 64 61 20 64 0008F         .ASCII  \d addr. desc.\                             ;
                              52 43 53 45 44 25 0009C P.AAF:  .ASCII  \%DESCR\                                    ;
45 4E 24 47 42 44 5C 4C 4C 41 43 47 42 44 27 000A2 P.AAG:  .ASCII  \'DBGCALL\<92>\DBG$NEXECUTE invalid val.\    ;
20 64 69 6C 61 76 6E 69 20 45 54 55 43 45 58 000B1                                                          ;
                                    2E 6C 61 76 000C0                                                          ;
                              2E 63 73 65 64 20 000C4         .ASCII  \ desc.\                                    ;
                                    46 45 52 25 000CA P.AAH:  .ASCII  \%REF\                                      ;
45 4E 24 47 42 44 5C 4C 4C 41 43 47 42 44 2C 000CE P.AAI:  .ASCII  \,DBGCALL\<92>\DBG$NEXECUTE_CALL, invali\    ;
6E 69 20 2C 4C 4C 41 43 5F 45 54 55 43 45 58 000DD                                                          ;
                                    69 6C 61 76 000EC                                                          ;
            63 73 65 64 20 2E 6C 61 76 20 64 000F0         .ASCII  \d val. desc\                               ;
                                    4C 41 56 25 000FB P.AAJ:  .ASCII  \%VAL\                                      ;
45 4E 24 47 42 44 5C 4C 4C 41 43 47 42 44 2C 000FF P.AAK:  .ASCII  \,DBGCALL\<92>\DBG$NEXECUTE_CALL, invali\    ;
6E 69 20 2C 4C 4C 41 43 5F 45 54 55 43 45 58 0010E                                                          ;
                                    69 6C 61 76 0011D                                                          ;
            63 73 65 64 20 2E 6C 61 76 20 64 00121         .ASCII  \d val. desc\                               ;

                                                           .PSECT  DBG$OWN,NOEXE,  PIC,2

                                   00000000 00000 SAVE_CALL_CONTEXT:
                                                           .LONG   0                                           ;

                                                           .PSECT  DBG$GLOBAL,NOEXE,  PIC,2

                                   00000000 00000 DBG$GL_CALL_CONTEXT::
                                                           .LONG   0                                           ;
                                         00 00004 DBG$GB_CALL_NORMAL_RET::
                                                           .BYTE   0                                           ;

                                                           .EXTRN  DBG$GET_MEMORY, DBG$GET_TEMPMEM
                                                           .EXTRN  DBG$MAKE_VMS_DESC
                                                           .EXTRN  DBG$NCOPY_DESC, DBG$NMATCH
                                                           .EXTRN  DBG$NNEXT_WORD, DBG$NPARSE_ADDRESS
                                                           .EXTRN  DBG$NPARSE_EXPRESSION
                                                           .EXTRN  DBG$NSAVE_STRING
                                                           .EXTRN  DBG$PRIM_TO_ADDR
                                                           .EXTRN  DBG$GB_TAKE_CMD
                                                           .EXTRN  DBG$PSEUDO_PROG
                                                           .EXTRN  DBG$RUNFRAME, DBG$GB_UNHANDLED_EXC

                                                           .PSECT  DBG$CODE,NOWRT,  SHR,  PIC,0

                                       0FFC 00000          .ENTRY  DBG$NEXECUTE_CALL, Save R2,R3,R4,R5,R6,R7,- ; 0231
                                                                   R8,R9,R10,R11                               ;
                53          04    AC    D0 00002          MOVL    VERB_NODE, R3                               ; 0288
                5B          01    A3    9A 00006          MOVZBL  1(R3), AST_FLAG                             ;
                55          04    A3    D0 0000A          MOVL    4(R3), ADVERB_NODE                          ; 0294
                            04    A5    DD 0000E          PUSHL   4(ADVERB_NODE)                              ; 0295
                52                6E    D0 00011          MOVL    VALUE_DESC, R2                              ; 0296
          79    8F          02    A2    91 00014          CMPB    2(R2), #121                                 ;
                                  2E    12 00019          BNEQ    2$                                          ;
                                  5E    DD 0001B          PUSHL   SP                                          ; 0299
                                  08    DD 0001D          PUSHL   #8                                          ;
                                  52    DD 0001F          PUSHL   R2                                          ;
```

```
                  00000000G  00             03  FB 00021       CALLS   #3, DBG$PRIM_TO_ADDR            ;
                             15             50  E8 00028       BLBS    R0, 1$                          ;
                                 00000000'  EF  9F 0002B       PUSHAB  P.AAA                           ; 0301
                                            01  DD 00031       PUSHL   #1                              ;
                                 00028362   8F  DD 00033       PUSHL   #164706                         ;
                  00000000G  00             03  FB 00039       CALLS   #3, LIB$SIGNAL                  ;
                             50             6E  D0 00040  1$:  MOVL    VALUE_DESC, R0                  ; 0303
                             5A        18   B0  D0 00043       MOVL    @24(R0), CALL_ADDRESS           ;
                                            20  11 00047       BRB     4$                              ; 0296
                         83  8F        02   A2  91 00049  2$:  CMPB    2(R2), #131                     ; 0312
                                            15  13 0004E       BEQL    3$                              ;
                                 00000000'  EF  9F 00050       PUSHAB  P.AAB                           ; 0314
                                            01  DD 00056       PUSHL   #1                              ;
                                 00028362   8F  DD 00058       PUSHL   #164706                         ;
                  00000000G  00             03  FB 0005E       CALLS   #3, LIB$SIGNAL                  ;
                             5A        18   A2  D0 00065  3$:  MOVL    24(R2), CALL_ADDRESS            ; 0316
              6A             01             00  0C 00069  4$:  PROBER  #0, #1, (CALL_ADDRESS)          ; 0322
                                            11  12 0006D       BNEQ    5$                              ;
                                            5A  DD 0006F       PUSHL   CALL_ADDRESS                    ; 0324
                                            01  DD 00071       PUSHL   #1                              ;
                                 000281E0   8F  DD 00073       PUSHL   #164320                         ;
                  00000000G  00             03  FB 00079       CALLS   #3, LIB$SIGNAL                  ;
                             7E             65  9A 00080  5$:  MOVZBL  (ADVERB_NODE), -(SP)            ; 0329
                                            6E  D6 00083       INCL    (SP)                            ;
                  00000000G  00             01  FB 00085       CALLS   #1, DBG$GET_MEMORY              ;
                             59             50  D0 0008C       MOVL    R0, ARG_LIST_PTR                ;
                                            58  D4 0008F       CLRL    CALARG_PERMEM                   ; 0330
                                            65  95 00091       TSTB    (ADVERB_NODE)                   ; 0331
                                            0D  13 00093       BEQL    6$                              ;
                             7E             65  9A 00095       MOVZBL  (ADVERB_NODE), -(SP)            ; 0333
                  00000000G  00             01  FB 00098       CALLS   #1, DBG$GET_MEMORY              ;
                             58             50  D0 0009F       MOVL    R0, CALARG_PERMEM               ;
                                            54  D4 000A2  6$:  CLRL    I                               ; 0338
                           6944             65  9A 000A4       MOVZBL  (ADVERB_NODE), (ARG_LIST_PTR)[I] ; 0339
                             57        08   A3  D0 000A8       MOVL    8(R3), NOUN_NODE                ; 0340
                                            03  12 000AC  7$:  BNEQ    8$                              ; 0343
                                          0164  31 000AE       BRW     33$                             ;
                             6E             67  D0 000B1  8$:  MOVL    (NOUN_NODE), VALUE_DESC         ; 0344
                             55        0C   A7  D0 000B4       MOVL    12(NOUN_NODE), BUF              ; 0345
                                            54  D6 000B8       INCL    I                               ; 0346
                                            56  D4 000BA       CLRL    R6                              ; 0349
 00000000'  EF           01  A5             05  29 000BC       CMPC3   #5, 1(BUF), P.AAC               ;
                                            02  12 000C5       BNEQ    9$                              ;
                                            56  D6 000C7       INCL    R6                              ;
                             01             56  D1 000C9  9$:  CMPL    R6, #1                          ;
                                            4F  12 000CC       BNEQ    13$                             ;
                             52             6E  D0 000CE       MOVL    VALUE_DESC, R2                  ; 0351
                         79  8F        02   A2  91 000D1       CMPB    2(R2), #121                     ;
                                            2F  12 000D6       BNEQ    11$                             ;
                                            5E  DD 000D8       PUSHL   SP                              ; 0354
                                            08  DD 000DA       PUSHL   #8                              ;
                                            52  DD 000DC       PUSHL   R2                              ;
                  00000000G  00             03  FB 000DE       CALLS   #3, DBG$PRIM_TO_ADDR            ;
                             15             50  E8 000E5       BLBS    R0, 10$                         ;
                                 00000000'  EF  9F 000E8       PUSHAB  P.AAD                           ; 0356
                                            01  DD 000EE       PUSHL   #1                              ;
                                 00028362   8F  DD 000F0       PUSHL   #164706                         ;
```

```
                               00000000G  00                03  FB  000F6         CALLS   #3, LIB$SIGNAL                             ;
                                          50                6E  D0  000FD  10$:   MOVL    VALUE_DESC, R0                             ; 0358
                                        6944        18      B0  D0  00100         MOVL    @24(R0), (ARG_LIST_PTR)[I]                 ;
                                                            7B  11  00105         BRB     19$                                        ; 0351
                                   83     8F        02      A2  91  00107  11$:   CMPB    2(R2), #131                                ; 0363
                                                            07  12  0010C         BNEQ    12$                                        ;
                                        6944        18      A2  D0  0010E         MOVL    24(R2), (ARG_LIST_PTR)[I]                  ; 0366
                                                            6D  11  00113         BRB     19$                                        ; 0363
                                             00000000'      EF  9F  00115  12$:   PUSHAB  P.AAE                                      ; 0371
                                                            4A  11  0011B         BRB     17$                                        ;
                                                            56  D4  0011D  13$:   CLRL    R6                                         ; 0378
                  00000000'  EF       01    A5              06  29  0011F         CMPC3   #6, 1(BUF), P.AAF                          ;
                                                            02  12  00128         BNEQ    14$                                        ;
                                                            56  D6  0012A         INCL    R6                                         ;
                                          01                56  D1  0012C  14$:   CMPL    R6, #1                                     ;
                                                            38  12  0012F         BNEQ    18$                                        ;
00000083   8F        00     BE            08                10  ED  00131         CMPZV   #16, #8, @VALUE_DESC, #131                 ; 0380
                                                            0C  13  0013B         BEQL    15$                                        ;
0000007A   8F        00     BE            08                10  ED  0013D         CMPZV   #16, #8, @VALUE_DESC, #122                 ; 0381
                                                            18  12  00147         BNEQ    16$                                        ;
                                                            5E  DD  00149  15$:   PUSHL   SP                                         ; 0384
                                                    04      AE  DD  0014B         PUSHL   VALUE_DESC                                 ;
                               00000000G  00                02  FB  0014E         CALLS   #2, DBG$NCOPY_DESC                         ;
                        6944              6E                14  C1  00155         ADDL3   #20, VALUE_DESC, (ARG_LIST_PTR)[I]         ; 0385
                                     FC A844                6E  D0  0015A         MOVL    VALUE_DESC, -4(CALARG_PERMEM)[I]           ; 0386
                                                            42  11  0015F         BRB     21$                                        ; 0380
                                             00000000'      EF  9F  00161  16$:   PUSHAB  P.AAG                                      ; 0390
                                                            42  11  00167  17$:   BRB     23$                                        ;
                               00000000'  EF        01      A5  D1  00169  18$:   CMPL    1(BUF), P.AAH                              ; 0394
                                                            3A  12  00171         BNEQ    24$                                        ;
                                          50                6E  D0  00173         MOVL    VALUE_DESC, R0                             ; 0396
                                   83     8F        02      A0  91  00176         CMPB    2(R0), #131                                ;
                                                            07  12  0017B         BNEQ    20$                                        ;
                                        6944        18      A0  D0  0017D         MOVL    24(R0), (ARG_LIST_PTR)[I]                  ; 0398
                                                            73  11  00182  19$:   BRB     29$                                        ;
                                   7A     8F        02      A0  91  00184  20$:   CMPB    2(R0), #122                                ; 0400
                                                            1A  12  00189         BNEQ    22$                                        ;
                                                  4001      8F  BB  0018B         PUSHR   #^M<R0,SP>                                 ; 0403
                               00000000G  00                02  FB  0018F         CALLS   #2, DBG$NCOPY_DESC                         ;
                                          50                6E  D0  00196         MOVL    VALUE_DESC, R0                             ; 0404
                                        6944        18      A0  D0  00199         MOVL    24(R0), (ARG_LIST_PTR)[I]                  ;
                                     FC A844                50  D0  0019E         MOVL    R0, -4(CALARG_PERMEM)[I]                   ; 0405
                                                            69  11  001A3  21$:   BRB     32$                                        ; 0400
                                             00000000'      EF  9F  001A5  22$:   PUSHAB  P.AAI                                      ; 0409
                                                            52  11  001AB  23$:   BRB     31$                                        ;
                               00000000'  EF        01      A5  D1  001AD  24$:   CMPL    1(BUF), P.AAJ                              ; 0413
                                                            57  12  001B5         BNEQ    32$                                        ;
                                          50                6E  D0  001B7         MOVL    VALUE_DESC, R0                             ; 0415
                                   7A     8F        02      A0  91  001BA         CMPB    2(R0), #122                                ;
                                                            38  12  001BF         BNEQ    30$                                        ;
                                        6944        18      B0  D0  001C1         MOVL    @24(R0), (ARG_LIST_PTR)[I]                 ; 0418
                                          01        16      A0  91  001C6         CMPB    22(R0), #1                                 ; 0419
                                                            06  13  001CA         BEQL    25$                                        ;
                                          22        16      A0  91  001CC         CMPB    22(R0), #34                                ; 0420
                                                            06  12  001D0         BNEQ    26$                                        ;
                                          20        14      A0  B1  001D2  25$:   CMPW    20(R0), #32                                ; 0423
                                                            10  11  001D6         BRB     28$                                        ;
```

```
                                    15        16   A0  91 001D8 26$:     CMPB    22(R0), #21                                  ; 0429
                                                   06  12 001DC          BNEQ    27$                                          ;
                                    08        14   A0  B1 001DE          CMPW    20(R0), #8                                   ; 0432
                                                   04  11 001E2          BRB     28$                                          ;
                                    04        14   A0  B1 001E4 27$:     CMPW    20(R0), #4                                   ; 0438
                                                   24  1B 001E8 28$:     BLEQU   32$                                          ;
                                         00028073  8F  DD 001EA          PUSHL   #163955                                      ; 0440
                          00000000G  00            01  FB 001F0          CALLS   #1, LIB$SIGNAL                               ;
                                                   15  11 001F7 29$:     BRB     32$                                          ; 0415
                                         00000000' EF  9F 001F9 30$:     PUSHAB  P.AAK                                        ; 0445
                                                   01  DD 001FF 31$:     PUSHL   #1                                           ;
                                         00028362  8F  DD 00201          PUSHL   #164706                                      ;
                          00000000G  00            03  FB 00207          CALLS   #3, LIB$SIGNAL                               ;
                                     57        08  A7  D0 0020E 32$:     MOVL    8(NOUN_NODE), NOUN_NODE                      ; 0451
                                                 FE97  31 00212          BRW     7$                                           ; 0341
                                                   1A  DD 00215 33$:     PUSHL   #26                                          ; 0461
                          00000000G  00            01  FB 00217          CALLS   #1, DBG$GET_MEMORY                           ;
                                     56            50  D0 0021E          MOVL    R0, SAVED_RUNFRAME                           ;
                       66 00000000G  00      0065  8F  28 00221          MOVC3   #101, DBG$RUNFRAME, (SAVED_RUNFRAME)         ; 0462
                          00000000G  00            56  D0 0022B          MOVL    SAVED_RUNFRAME, DBG$RUNFRAME                 ; 0463
                          00000000G  00 00000000G  00  9E 00232          MOVAB   DBG$PSEUDO_PROG, DBG$RUNFRAME+64             ; 0464
                          00000000G  00 03C00000   8F  D0 0023D          MOVL    #62914560, DBG$RUNFRAME+68                   ; 0465
        0000*  8F               00         6E      00  2C 00248          MOVC5   #0, (SP), #0, #<<DBG$RUNFRAME+101>--         ; 0468
                                         00000000G 00     0024F                  <DBG$RUNFRAME+72>>, DBG$RUNFRAME+72           ;
                                     0F            5B  E9 00254          BLBC    AST_FLAG, 34$                                ; 0469
               50         48    A6           01    05  EF 00257          EXTZV   #5, #1, 72(SAVED_RUNFRAME), R0               ; 0471
   00000000G   00               01           05    50  F0 0025D          INSV    R0, #5, #1, DBG$RUNFRAME+72                  ;
                          00000000G  00            59  7D 00266 34$:     MOVQ    ARG_LIST_PTR, DBG$RUNFRAME+78                ; 0472
                          00000000G  00            58  D0 0026D          MOVL    CALARG_PERMEM, DBG$RUNFRAME+97               ; 0474
                          00000000G  00 00000000'  EF  D0 00274          MOVL    SAVE_CALL_CONTEXT, DBG$RUNFRAME+8            ; 0475
                                     50            09  D0 0027F          MOVL    #9, I                                        ; 0488
                          00000000G0040 00000000G0040  90 00282 35$:     MOVB    DBG$GB_UNHANDLED_EXC-1[I], -                 ; 0489
                                                                                 DBG$GB_UNHANDLED_EXC[I]                      ;
                                     F0            50  F5 0028F          SOBGTR  I, 35$                                       ;
                                         00000000G 00  94 00292          CLRB    DBG$GB_UNHANDLED_EXC                         ; 0490
                          00000000'  EF            01  90 00298          MOVB    #1, DBG$GB_CALL_NORMAL_RET                   ; 0496
                                         00000000G 00  94 0029F          CLRB    DBG$GB_TAKE_CMD                              ; 0497
                                     50            01  D0 002A5          MOVL    #1, R0                                       ; 0498
                                                       04 002A8          RET                                                  ; 0499

; Routine Size:  681 bytes,    Routine Base:  DBG$CODE + 0000
```

```
;  370          0500  1 GLOBAL ROUTINE DBG$NPARSE_CALL(INPUT_DESC, VERB_NODE, MESSAGE_VECT) =
;  371          0501  1 !
;  372          0502  1 ! FUNCTION
;  373          0503  1 !       Parse network for the CALL command.  The parsing method used is
;  374          0504  1 !       that of ATN's.  This network constructs a command execution tree to
;  375          0505  1 !       be executed by DBG$NEXECUTE_CALL.
;  376          0506  1 !
;  377          0507  1 !       CALL addr-exp(addr-exp, %ADDR addr-exp, %REF lang-exp, %VAL lang-exp,
;  378          0508  1 !                     %DESCR lang-exp, ...)
;  379          0509  1 !
;  380          0510  1 ! INPUTS
;  381          0511  1 !       INPUT_DESC      - A longword containing the address of a standard
;  382          0512  1 !                         string descriptor which reflects the input string.
;  383          0513  1 !
;  384          0514  1 !       VERB_NODE       - A longword containing the address of the verb
;  385          0515  1 !                         node of the command execution tree. (CALL)
;  386          0516  1 !
;  387          0517  1 !       MESSAGE_VECT    - The address of a longword to contain the address
;  388          0518  1 !                         of a message argument vector.
;  389          0519  1 !
;  390          0520  1 ! OUTPUTS
;  391          0521  1 !       STS$K_SUCCESS (1)       - Success.  Input parsed and execution tree
;  392          0522  1 !                                 constructed.
;  393          0523  1 !
;  394          0524  1 !       STS$K_SEVERE  (4)       - Failure.  Tree not constructed.  Message
;  395          0525  1 !                                 vector constructed.
;  396          0526  1 !
;  397          0527  2   BEGIN
;  398          0528  2
;  399          0529  2   MAP
;  400          0530  2       INPUT_DESC: REF BLOCK[,BYTE],   ! Pointer to Input Descriptor
;  401          0531  2       VERB_NODE: REF DBG$VERB_NODE;   ! Pointer to Command Verb Node
;  402          0532  2
;  403          0533  2   BIND
;  404          0534  2       DBG$CS_AST          = UPLIT BYTE (%ASCIC 'AST'),
;  405          0535  2       DBG$CS_NOAST        = UPLIT BYTE (%ASCIC 'NOAST'),
;  406          0536  2       DBG$CS_COMMA        = UPLIT BYTE(1, DBG$K_COMMA),
;  407          0537  2       DBG$CS_CR           = UPLIT BYTE(1, DBG$K_CAR_RETURN),
;  408          0538  2       DBG$CS_LEFT_PAREN = UPLIT BYTE(1, DBG$K_LEFT_PARENTHESIS),
;  409          0539  2       DBG$CS_RGHT_PAREN = UPLIT BYTE(1, DBG$K_RIGHT_PARENTHESIS),
;  410          0540  2       DBG$CS_SLASH        = UPLIT BYTE(1, '/');
;  411          0541  2
;  412          0542  2   LOCAL
;  413          0543  2       ADVERB_NODE: REF DBG$ADVERB_NODE,! Pointer to Command Adverb Node
;  414          0544  2       AST_FLAG,                       ! TRUE for CALL/AST
;  415          0545  2       BUF: REF VECTOR[,BYTE],         ! ASCIC string
;  416          0546  2       NOUN_NODE: REF DBG$NOUN_NODE,   ! Pointer to Command Noun Node
;  417          0547  2       LINK,                           ! Pointer to next noun node
;  418          0548  2       SAVE_INPUT_DESC: DBG$STG_DESC,  ! Save the Input Descriptor
;  419          0549  2       STATUS;                         ! Returned status
;  420          0550  2
;  421          0551  2
;  422          0552  2   ! Check for /AST or /NOAST, which controls whether we will re-enable
;  423          0553  2   ! ASTs while the user program that is CALLed is running.
;  424          0554  2   ! If we see /AST then we set AST_FLAG to TRUE, if we
;  425          0555  2   ! see /NOAST then we set AST_FLAG to FALSE.
;  426          0556  2   ! AST_FLAG is initially TRUE, meaning that the default is /AST.
```

```
;  427          0557  2      ! This information is put in the VERB_COMPOSITE field and looked
;  428          0558  2      ! at in DBG$NEXECUTE_CALL.
;  429          0559  2      !
;  430          0560  2      AST_FLAG = TRUE;
;  431          0561  2      WHILE DBG$NMATCH(.INPUT_DESC, DBG$CS_SLASH, 1) DO
;  432          0562  2          BEGIN
;  433          0563  3          SELECTONE TRUE OF
;  434          0564  3              SET
;  435          0565  3              [DBG$NMATCH(.INPUT_DESC, DBG$CS_AST, 1)]:
;  436          0566  3                  AST_FLAG = TRUE;
;  437          0567  3              [DBG$NMATCH(.INPUT_DESC, DBG$CS_NOAST, 1)]:
;  438          0568  3                  AST_FLAG = FALSE;
;  439          0569  3              [OTHERWISE]:
;  440          0570  3                  SIGNAL(DBG$_CMDSYNERR, 1, DBG$NNEXT_WORD(.INPUT_DESC));
;  441          0571  3              TES;
;  442          0572  2          END;
;  443          0573  2      VERB_NODE[DBG$B_VERB_COMPOSITE] = .AST_FLAG;
;  444          0574  2
;  445          0575  2      ! Signal an error if no parameters are present at all.
;  446          0576  2      !
;  447          0577  2      IF DBG$NMATCH(.INPUT_DESC, DBG$CS_CR, 1) THEN SIGNAL(DBG$_NEEDMORE);
;  448          0578  2
;  449          0579  2
;  450          0580  2      ! Pick up the routine address to call.
;  451          0581  2      !
;  452          0582  2      ADVERB_NODE = DBG$GET_TEMPMEM(DBG$K_ADVERB_NODE_SIZE);
;  453          0583  2      VERB_NODE[DBG$L_VERB_ADVERB_PTR] = .ADVERB_NODE;
;  454          0584  2      DBG$GL_CALL_CONTEXT = .DBG$RUNFRAME[DBG$L_USER_R1];
;  455          0585  2      STATUS = DBG$NPARSE_ADDRESS(.INPUT_DESC, ADVERB_NODE[DBG$L_ADVERB_VALUE],
;  456          0586  2                  DBG$K_DEFAULT, TOKEN$K_TERM_OPEN);
;  457          0587  2      SAVE_CALL_CONTEXT = DBG$GL_CALL_CONTEXT;
;  458          0588  2
;  459          0589  2
;  460          0590  2      ! Initialize the argument count to zero.
;  461          0591  2      !
;  462          0592  2      ADVERB_NODE[DBG$B_ADVERB_LITERAL] = 0;
;  463          0593  2
;  464          0594  2
;  465          0595  2      ! Check for the returned status.  If STS$K_WARNING is returned, then the
;  466          0596  2      ! CALL command must have arguments.
;  467          0597  2      !
;  468          0598  2      IF .STATUS NEQ STS$K_SUCCESS
;  469          0599  2      THEN
;  470          0600  3          BEGIN
;  471          0601  3
;  472          0602  3
;  473          0603  3          ! Check for the valid syntax ('(' before the arguments).
;  474          0604  3          !
;  475          0605  3          IF DBG$NMATCH(.INPUT_DESC, DBG$CS_LEFT_PAREN, 1)
;  476          0606  3          THEN
;  477          0607  4              BEGIN
;  478          0608  4              LINK = VERB_NODE[DBG$L_VERB_OBJECT_PTR];
;  479          0609  4              WHILE TRUE DO
;  480          0610  5                  BEGIN
;  481          0611  5                  LOCAL
;  482          0612  5                      COUNT;
;  483          0613  5
```

```
;  484          0614  5          ADVERB_NODE[DBG$B_ADVERB_LITERAL] =
;  485          0615  5              .ADVERB_NODE[DBG$B_ADVERB_LITERAL] + 1;
;  486          0616  5          CH$MOVE(8, .INPUT_DESC, SAVE_INPUT_DESC);
;  487          0617  5          BUF = .SAVE_INPUT_DESC[DSC$A_POINTER];
;  488          0618  5          COUNT = 0;
;  489          0619  5          WHILE .BUF[0] EQL %C' ' DO
;  490          0620  6              BEGIN
;  491          0621  6              BUF = .BUF + 1;
;  492          0622  6              COUNT = .COUNT + 1;
;  493          0623  5              END;
;  494          0624  5          SAVE_INPUT_DESC[DSC$W_LENGTH]
;  495          0625  5              = .SAVE_INPUT_DESC[DSC$W_LENGTH] - .COUNT;
;  496          0626  5          SAVE_INPUT_DESC[DSC$A_POINTER] = .BUF;
;  497          0627  5
;  498          0628  5          NOUN_NODE = DBG$GET_TEMPMEM(DBG$K_NOUN_NODE_SIZE);
;  499          0629  5          .LINK = .NOUN_NODE;
;  500          0630  5          LINK = NOUN_NODE[DBG$L_NOUN_LINK];
;  501          0631  5          IF NOT DBG$NSAVE_STRING(.INPUT_DESC,
;  502          0632  5                  NOUN_NODE[DBG$L_NOUN_VALUE2], .MESSAGE_VECT)
;  503          0633  5          THEN
;  504          0634  5              RETURN STS$K_SEVERE;
;  505          0635  5          BUF = .NOUN_NODE[DBG$L_NOUN_VALUE2];
;  506          0636  5          SELECTONE TRUE OF
;  507          0637  5              SET
;  508          0638  5              [CH$EQL(5, BUF[1], 5, UPLIT BYTE('%ADDR'))]:
;  509          0639  6                  BEGIN
;  510          0640  6                  INPUT_DESC[DSC$W_LENGTH]
;  511          0641  6                      = .SAVE_INPUT_DESC[DSC$W_LENGTH] - 5;
;  512          0642  6                  INPUT_DESC[DSC$A_POINTER]
;  513          0643  6                      = .SAVE_INPUT_DESC[DSC$A_POINTER] + 5;
;  514          0644  6                  STATUS = DBG$NPARSE_ADDRESS(.INPUT_DESC,
;  515          0645  6                                              NOUN_NODE[DBG$L_NOUN_VALUE],
;  516          0646  6                                              DBG$K_DEFAULT,
;  517          0647  6                                              TOKEN$K_TERM_COMPAREN);
;  518          0648  5                  END;
;  519          0649  5
;  520          0650  5              [CH$EQL(6, BUF[1], 6, UPLIT BYTE('%DESCR'))]:
;  521          0651  6                  BEGIN
;  522          0652  6                  INPUT_DESC[DSC$W_LENGTH]
;  523          0653  6                      = .SAVE_INPUT_DESC[DSC$W_LENGTH] - 6;
;  524          0654  6                  INPUT_DESC[DSC$A_POINTER]
;  525          0655  6                      = .SAVE_INPUT_DESC[DSC$A_POINTER] + 6;
;  526          0656  6                  STATUS = DBG$NPARSE_EXPRESSION(.INPUT_DESC,
;  527          0657  6                                              DBG$K_DEFAULT,
;  528          0658  6                                              NOUN_NODE[DBG$L_NOUN_VALUE],
;  529          0659  6                                              TOKEN$K_TERM_COMPAREN);
;  530          0660  5                  END;
;  531          0661  5
;  532          0662  5              [CH$EQL(4, BUF[1], 4, UPLIT BYTE('%REF'))]:
;  533          0663  6                  BEGIN
;  534          0664  6                  INPUT_DESC[DSC$W_LENGTH]
;  535          0665  6                      = .SAVE_INPUT_DESC[DSC$W_LENGTH] - 4;
;  536          0666  6                  INPUT_DESC[DSC$A_POINTER]
;  537          0667  6                      = .SAVE_INPUT_DESC[DSC$A_POINTER] + 4;
;  538          0668  6                  STATUS = DBG$NPARSE_EXPRESSION(.INPUT_DESC,
;  539          0669  6                                              DBG$K_DEFAULT,
;  540          0670  6                                              NOUN_NODE[DBG$L_NOUN_VALUE],
```

```
;   541     0671  6                                   TOKEN$K_TERM_COMPAREN);
;   542     0672  5                           END;
;   543     0673  5
;   544     0674  5                       [CH$EQL(4, BUF[1], 4, UPLIT BYTE('%VAL'))]:
;   545     0675  6                           BEGIN
;   546     0676  6                           INPUT_DESC[DSC$W_LENGTH]
;   547     0677  6                               = .SAVE_INPUT_DESC[DSC$W_LENGTH] - 4;
;   548     0678  6                           INPUT_DESC[DSC$A_POINTER]
;   549     0679  6                               = .SAVE_INPUT_DESC[DSC$A_POINTER] + 4;
;   550     0680  6                           STATUS = DBG$NPARSE_EXPRESSION(.INPUT_DESC,
;   551     0681  6                                                          DBG$K_DEFAULT,
;   552     0682  6                                                          NOUN_NODE[DBG$L_NOUN_VALUE],
;   553     0683  6                                                          TOKEN$K_TERM_COMPAREN);
;   554     0684  5                           END;
;   555     0685  5
;   556     0686  5                       [OTHERWISE]:
;   557     0687  6                           BEGIN
;   558     0688  6                           NOUN_NODE[DBG$L_NOUN_VALUE2] = UPLIT BYTE(%ASCIC '%ADDR');
;   559     0689  6                           CH$MOVE(8, SAVE_INPUT_DESC, .INPUT_DESC);
;   560     0690  6                           STATUS = DBG$NPARSE_ADDRESS(.INPUT_DESC,
;   561     0691  6                                                       NOUN_NODE[DBG$L_NOUN_VALUE],
;   562     0692  6                                                       DBG$K_DEFAULT,
;   563     0693  6                                                       TOKEN$K_TERM_COMPAREN);
;   564     0694  5                           END;
;   565     0695  5
;   566     0696  5                       TES;
;   567     0697  5
;   568     0698  5                   IF .STATUS EQL STS$K_SUCCESS THEN SIGNAL(DBG$_NEEDMORE);
;   569     0699  5                   IF DBG$NMATCH(.INPUT_DESC, DBG$CS_RGHT_PAREN, 1) THEN EXITLOOP;
;   570     0700  5                   IF NOT DBG$NMATCH(.INPUT_DESC, DBG$CS_COMMA, 1)
;   571     0701  5                   THEN
;   572     0702  5                       SIGNAL(DBG$_CMDSYNERR, 1, DBG$NNEXT_WORD(.INPUT_DESC));
;   573     0703  5
;   574     0704  4                       END;                    ! End of WHILE parsing (...) loop.
;   575     0705  4
;   576     0706  4                   END
;   577     0707  4
;   578     0708  3               ELSE
;   579     0709  3                   SIGNAL(DBG$_CMDSYNERR, 1, DBG$NNEXT_WORD(.INPUT_DESC));
;   580     0710  3
;   581     0711  2               END;
;   582     0712  2
;   583     0713  2           IF NOT DBG$NMATCH(.INPUT_DESC, DBG$CS_CR, 1)
;   584     0714  2           THEN
;   585     0715  2               SIGNAL(DBG$_CMDSYNERR, 1, DBG$NNEXT_WORD(.INPUT_DESC));
;   586     0716  2
;   587     0717  2           RETURN STS$K_SUCCESS;
;   588     0718  2
;   589     0719  1           END;


                                                        .PSECT  DBG$PLIT,NOWRT,  SHR,  PIC,0

                                    54  53  41  03  0012C P.AAL:  .ASCII  <3>\AST\
                            54  53  41  4F  4E  05  00130 P.AAM:  .ASCII  <5>\NOAST\
                                                2C  01  00136 P.AAN:  .BYTE   1, 44
```

```
                               0D  01  00138 P.AAO:   .BYTE   1, 13                                    ;
                               28  01  0013A P.AAP:   .BYTE   1, 40                                    ;
                               29  01  0013C P.AAQ:   .BYTE   1, 41                                    ;
                                   01  0013E P.AAR:   .BYTE   1                                        ;
                                   2F  0013F          .ASCII  \/\                                      ;
               52  44  44  41  25  00140 P.AAS:   .ASCII  \%ADDR\                                  ;
           52  43  53  45  44  25  00145 P.AAT:   .ASCII  \%DESCR\                                 ;
                   46  45  52  25  0014B P.AAU:   .ASCII  \%REF\                                   ;
                   4C  41  56  25  0014F P.AAV:   .ASCII  \%VAL\                                   ;
           52  44  44  41  25  05  00153 P.AAW:   .ASCII  <5>\%ADDR\                               ;

                                         DBG$CS_AST=            P.AAL
                                         DBG$CS_NOAST=          P.AAM
                                         DBG$CS_COMMA=          P.AAN
                                         DBG$CS_CR=             P.AAO
                                         DBG$CS_LEFT_PAREN=     P.AAP
                                         DBG$CS_RGHT_PAREN=     P.AAQ
                                         DBG$CS_SLASH=          P.AAR

                                                  .PSECT  DBG$CODE,NOWRT,  SHR,  PIC,0

                                  0FFC 00000      .ENTRY  DBG$NPARSE_CALL, Save R2,R3,R4,R5,R6,R7,R8,-; 0500
                                                          R9,R10,R11                                 ;
            5E                 0C  C2 00002       SUBL2   #12, SP                                    ;
            52                 01  D0 00005       MOVL    #1, AST_FLAG                               ; 0560
            57         04      AC  D0 00008       MOVL    INPUT_DESC, R7                             ; 0561
                               01  DD 0000C 1$:   PUSHL   #1                                         ;
                  00000000'    EF  9F 0000E       PUSHAB  DBG$CS_SLASH                               ;
                               57  DD 00014       PUSHL   R7                                         ;
   00000000G  00               03  FB 00016       CALLS   #3, DBG$NMATCH                             ;
              51               50  E9 0001D       BLBC    R0, 4$                                     ;
                               01  DD 00020       PUSHL   #1                                         ; 0565
                  00000000'    EF  9F 00022       PUSHAB  DBG$CS_AST                                 ;
                               57  DD 00028       PUSHL   R7                                         ;
   00000000G  00               03  FB 0002A       CALLS   #3, DBG$NMATCH                             ;
              01               50  D1 00031       CMPL    R0, #1                                     ;
                               05  12 00034       BNEQ    2$                                         ;
              52               01  D0 00036       MOVL    #1, AST_FLAG                               ; 0566
                               D1  11 00039       BRB     1$                                         ;
                               01  DD 0003B 2$:   PUSHL   #1                                         ; 0567
                  00000000'    EF  9F 0003D       PUSHAB  DBG$CS_NOAST                               ;
                               57  DD 00043       PUSHL   R7                                         ;
   00000000G  00               03  FB 00045       CALLS   #3, DBG$NMATCH                             ;
              01               50  D1 0004C       CMPL    R0, #1                                     ;
                               04  12 0004F       BNEQ    3$                                         ;
                               52  D4 00051       CLRL    AST_FLAG                                   ; 0568
                               B7  11 00053       BRB     1$                                         ;
                               57  DD 00055 3$:   PUSHL   R7                                         ; 0570
   00000000G  00               01  FB 00057       CALLS   #1, DBG$NNEXT_WORD                         ;
                               50  DD 0005E       PUSHL   R0                                         ;
                               01  DD 00060       PUSHL   #1                                         ;
                  00028EB8     8F  DD 00062       PUSHL   #167608                                    ;
   00000000G  00               03  FB 00068       CALLS   #3, LIB$SIGNAL                             ;
                               9B  11 0006F       BRB     1$                                         ; 0561
              56       08      AC  D0 00071 4$:   MOVL    VERB_NODE, R6                              ; 0573
          01  A6               52  90 00075       MOVB    AST_FLAG, 1(R6)                            ;
```

```
                                              01  DD  00079          PUSHL   #1                                      ; 0577
                           00000000'          EF  9F  0007B          PUSHAB  DBG$CS_CR                               ;
                                              57  DD  00081          PUSHL   R7                                      ;
               00000000G  00                  03  FB  00083          CALLS   #3, DBG$NMATCH                          ;
                          0D                  50  E9  0008A          BLBC    R0, 5$                                  ;
                           000280D0           8F  DD  0008D          PUSHL   #164048                                 ;
               00000000G  00                  01  FB  00093          CALLS   #1, LIB$SIGNAL                          ;
                                              03  DD  0009A  5$:     PUSHL   #3                                      ; 0582
               00000000G  00                  01  FB  0009C          CALLS   #1, DBG$GET_TEMPMEM                     ;
                          5A                  50  D0  000A3          MOVL    R0, ADVERB_NODE                         ;
                      04  A6                  5A  D0  000A6          MOVL    ADVERB_NODE, 4(R6)                      ; 0583
               00000000'  EF  00000000G       00  D0  000AA          MOVL    DBG$RUNFRAME+8, DBG$GL_CALL_CONTEXT     ; 0584
                                              0B  DD  000B5          PUSHL   #11                                     ; 0585
                                              01  DD  000B7          PUSHL   #1                                      ;
                                        04    AA  9F  000B9          PUSHAB  4(ADVERB_NODE)                          ;
                                              57  DD  000BC          PUSHL   R7                                      ;
               00000000G  00                  04  FB  000BE          CALLS   #4, DBG$NPARSE_ADDRESS                  ;
                          5B                  50  D0  000C5          MOVL    R0, STATUS                              ;
               00000000'  EF  00000000'       EF  9E  000C8          MOVAB   DBG$GL_CALL_CONTEXT, SAVE_CALL_CONTEXT  ; 0587
                                              6A  94  000D3          CLRB    (ADVERB_NODE)                           ; 0592
                          01                  5B  D1  000D5          CMPL    STATUS, #1                              ; 0598
                                              03  12  000D8          BNEQ    6$                                      ;
                                            0162  31  000DA          BRW     24$                                     ;
                                              01  DD  000DD  6$:     PUSHL   #1                                      ; 0605
                           00000000'          EF  9F  000DF          PUSHAB  DBG$CS_LEFT_PAREN                       ;
                                              57  DD  000E5          PUSHL   R7                                      ;
               00000000G  00                  03  FB  000E7          CALLS   #3, DBG$NMATCH                          ;
                          03                  50  E8  000EE          BLBS    R0, 7$                                  ;
                                            0131  31  000F1          BRW     23$                                     ;
                          56                  08  C0  000F4  7$:     ADDL2   #8, LINK                                ; 0608
                                              6A  96  000F7  8$:     INCB    (ADVERB_NODE)                           ; 0615
          6E              67                  08  28  000F9          MOVC3   #8, (R7), SAVE_INPUT_DESC               ; 0616
                          59            04    AE  D0  000FD          MOVL    SAVE_INPUT_DESC+4, BUF                  ; 0617
                                              50  D4  00101          CLRL    COUNT                                   ; 0618
                          20                  69  91  00103  9$:     CMPB    (BUF), #32                              ; 0619
                                              06  12  00106          BNEQ    10$                                     ;
                                              59  D6  00108          INCL    BUF                                     ; 0621
                                              50  D6  0010A          INCL    COUNT                                   ; 0622
                                              F5  11  0010C          BRB     9$                                      ; 0619
                          6E                  50  A2  0010E  10$:    SUBW2   COUNT, SAVE_INPUT_DESC                  ; 0625
                      04  AE                  59  D0  00111          MOVL    BUF, SAVE_INPUT_DESC+4                  ; 0626
                                              04  DD  00115          PUSHL   #4                                      ; 0628
               00000000G  00                  01  FB  00117          CALLS   #1, DBG$GET_TEMPMEM                     ;
                          58                  50  D0  0011E          MOVL    R0, NOUN_NODE                           ;
                          66                  58  D0  00121          MOVL    NOUN_NODE, (LINK)                       ; 0629
                          56            08    A8  9E  00124          MOVAB   8(R8), LINK                             ; 0630
                                        0C    AC  DD  00128          PUSHL   MESSAGE_VECT                            ; 0632
                                        0C    A8  9F  0012B          PUSHAB  12(NOUN_NODE)                           ;
                                              57  DD  0012E          PUSHL   R7                                      ;
               00000000G  00                  03  FB  00130          CALLS   #3, DBG$NSAVE_STRING                    ;
                          04                  50  E8  00137          BLBS    R0, 11$                                 ;
                          50                  04  D0  0013A          MOVL    #4, R0                                  ; 0634
                                                  04  0013D          RET                                             ;
                          59            0C    A8  D0  0013E  11$:    MOVL    12(NOUN_NODE), BUF                      ; 0635
                                              54  D4  00142          CLRL    R4                                      ; 0638
00000000'  EF         01  A9                  05  29  00144          CMPC3   #5, 1(BUF), P.AAS                       ;
                                              02  12  0014D          BNEQ    12$                                     ;
```

```
                                           54  D6 0014F          INCL    R4                                   ;
                          01               54  D1 00151 12$:     CMPL    R4, #1                               ;
                                           0C  12 00154          BNEQ    13$                                  ;
              67          6E               05  A3 00156          SUBW3   #5, SAVE_INPUT_DESC, (R7)            ; 0641
        04    A7     04   AE               05  C1 0015A          ADDL3   #5, SAVE_INPUT_DESC+4, 4(R7)         ; 0643
                                           5B  11 00160          BRB     19$                                  ; 0645
                                           54  D4 00162 13$:     CLRL    R4                                   ; 0650
  00000000'   EF     01   A9               06  29 00164          CMPC3   #6, 1(BUF), P.AAT                    ;
                                           02  12 0016D          BNEQ    14$                                  ;
                                           54  D6 0016F          INCL    R4                                   ;
                          01               54  D1 00171 14$:     CMPL    R4, #1                               ;
                                           0C  12 00174          BNEQ    15$                                  ;
              67          6E               06  A3 00176          SUBW3   #6, SAVE_INPUT_DESC, (R7)            ; 0653
        04    A7     04   AE               06  C1 0017A          ADDL3   #6, SAVE_INPUT_DESC+4, 4(R7)         ; 0655
                                           1E  11 00180          BRB     17$                                  ; 0658
           00000000'      EF          01   A9  D1 00182 15$:     CMPL    1(BUF), P.AAU                        ; 0662
                                           0A  13 0018A          BEQL    16$                                  ;
           00000000'      EF          01   A9  D1 0018C          CMPL    1(BUF), P.AAV                        ; 0674
                                           1B  12 00194          BNEQ    18$                                  ;
              67          6E               04  A3 00196 16$:     SUBW3   #4, SAVE_INPUT_DESC, (R7)            ; 0677
        04    A7     04   AE               04  C1 0019A          ADDL3   #4, SAVE_INPUT_DESC+4, 4(R7)         ; 0679
                                           0C  DD 001A0 17$:     PUSHL   #12                                  ; 0682
                                           58  DD 001A2          PUSHL   NOUN_NODE                            ;
                                           01  DD 001A4          PUSHL   #1                                   ;
                                           57  DD 001A6          PUSHL   R7                                   ;
           00000000G      00               04  FB 001A8          CALLS   #4, DBG$NPARSE_EXPRESSION            ;
                                           1A  11 001AF          BRB     20$                                  ;
                     0C   A8  00000000'    EF  9E 001B1 18$:     MOVAB   P.AAW, 12(NOUN_NODE)                 ; 0688
              67          6E               08  28 001B9          MOVC3   #8, SAVE_INPUT_DESC, (R7)            ; 0689
                                           0C  DD 001BD 19$:     PUSHL   #12                                  ; 0691
                                           01  DD 001BF          PUSHL   #1                                   ;
                          7E               57  7D 001C1          MOVQ    R7, -(SP)                            ;
           00000000G      00               04  FB 001C4          CALLS   #4, DBG$NPARSE_ADDRESS               ;
                          5B               50  D0 001CB 20$:     MOVL    R0, STATUS                           ;
                          01               5B  D1 001CE          CMPL    STATUS, #1                           ; 0698
                                           0D  12 001D1          BNEQ    21$                                  ;
                              000280D0     8F  DD 001D3          PUSHL   #164048                              ;
           00000000G      00               01  FB 001D9          CALLS   #1, LIB$SIGNAL                       ;
                                           01  DD 001E0 21$:     PUSHL   #1                                   ; 0699
                              00000000'    EF  9F 001E2          PUSHAB  DBG$CS_RGHT_PAREN                    ;
                                           57  DD 001E8          PUSHL   R7                                   ;
           00000000G      00               03  FB 001EA          CALLS   #3, DBG$NMATCH                       ;
                          4B               50  E8 001F1          BLBS    R0, 24$                              ;
                                           01  DD 001F4          PUSHL   #1                                   ; 0700
                              00000000'    EF  9F 001F6          PUSHAB  DBG$CS_COMMA                         ;
                                           57  DD 001FC          PUSHL   R7                                   ;
           00000000G      00               03  FB 001FE          CALLS   #3, DBG$NMATCH                       ;
                          1A               50  E8 00205          BLBS    R0, 22$                              ;
                                           57  DD 00208          PUSHL   R7                                   ; 0702
           00000000G      00               01  FB 0020A          CALLS   #1, DBG$NNEXT_WORD                   ;
                                           50  DD 00211          PUSHL   R0                                   ;
                                           01  DD 00213          PUSHL   #1                                   ;
                              00028EB8     8F  DD 00215          PUSHL   #167608                              ;
           00000000G      00               03  FB 0021B          CALLS   #3, LIB$SIGNAL                       ;
                                         FED2  31 00222 22$:     BRW     8$                                   ; 0609
                                           57  DD 00225 23$:     PUSHL   R7                                   ; 0709
           00000000G      00               01  FB 00227          CALLS   #1, DBG$NNEXT_WORD                   ;
```

```
                                        50  DD 0022E          PUSHL   R0                                  ;
                                        01  DD 00230          PUSHL   #1                                  ;
                          00028EB8      8F  DD 00232          PUSHL   #167608                             ;
             00000000G  00              03  FB 00238          CALLS   #3, LIB$SIGNAL                      ;
                                        01  DD 0023F  24$:    PUSHL   #1                                  ; 0713
                          00000000'     EF  9F 00241          PUSHAB  DBG$CS_CR                           ;
                                        57  DD 00247          PUSHL   R7                                  ;
             00000000G  00              03  FB 00249          CALLS   #3, DBG$NMATCH                      ;
                        1A              50  E8 00250          BLBS    R0, 25$                             ;
                                        57  DD 00253          PUSHL   R7                                  ; 0715
             00000000G  00              01  FB 00255          CALLS   #1, DBG$NNEXT_WORD                  ;
                                        50  DD 0025C          PUSHL   R0                                  ;
                                        01  DD 0025E          PUSHL   #1                                  ;
                          00028EB8      8F  DD 00260          PUSHL   #167608                             ;
             00000000G  00              03  FB 00266          CALLS   #3, LIB$SIGNAL                      ;
                        50              01  D0 0026D  25$:    MOVL    #1, R0                              ; 0717
                                            04 00270          RET                                         ; 0719

; Routine Size:  625 bytes,    Routine Base:  DBG$CODE + 02A9


;  590          0720  1
;  591          0721  0 END ELUDOM



                                                              .EXTRN  LIB$SIGNAL

;                               PSECT SUMMARY
;
;      Name                    Bytes                        Attributes
;
;  DBG$GLOBAL                      5  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;  DBG$OWN                         4  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;  DBG$PLIT                      345  NOVEC,NOWRT,  RD , EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(0)
;  DBG$CODE                     1306  NOVEC,NOWRT,  RD , EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(0)


;                          Library Statistics
;
;                                            -------- Symbols --------       Pages       Processing
;      File                                  Total    Loaded   Percent       Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1             18619         8         0         1000       00:01.8
;  _$255$DUA28:[DEBUG.OBJ]STRUCDEF.L32;1        32         0         0            7       00:00.1
;  _$255$DUA28:[DEBUG.OBJ]DBGLIB.L32;1        1545        72         4           97       00:01.9
;  _$255$DUA28:[DEBUG.OBJ]DSTRECRDS.L32;1
;                                              418         0         0           31       00:00.3
;  _$255$DUA28:[DEBUG.OBJ]DBGMSG.L32;1         386        15         3           22       00:00.3
```

```
;                                       COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:DBGCALL/OBJ=OBJ$:DBGCALL MSRC$:DBGCALL/UPDATE=(ENH$:DBGCALL)

; Size:          1306 code + 354 data bytes
; Run Time:        00:24.7
; Elapsed Time:    01:32.0
; Lines/CPU Min:    1754
; Lexemes/CPU-Min: 14056
; Memory Used:  234 pages
; Compilation Complete
```

DBGCALL
LIS

DBGCVTDX
LIS